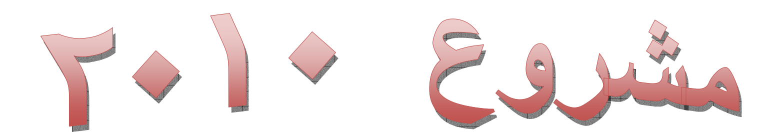
برنامج متابعة حضور الطلاب للسكاشن



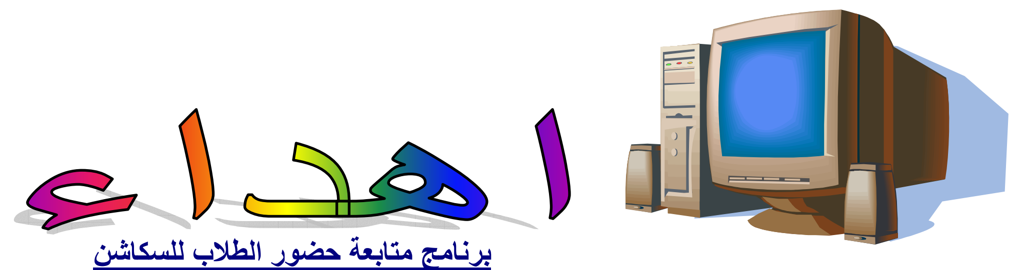






أ. أحمد فتحي محمد سليمان الشريف





<u>برنامج متابعة حضور الطلاب للسكاشن</u>

<u>مجموعه رقم ٤</u>

بطاقـــــة الشـــــكر

لابد لنا ونحن نخطو خطواتنا الأخيرة في الحياة الجامعية من وقفة نعود إلى أعوام قضيناها بالمعهد مع أساتذتنا الكرام الذين قـدموا لنـا الكثيـر بـاذلين بـذلك جهـودا كبيـرة فـى بنـاء جيـل الغـد لتبعـث الأمـة مـن جديـد وقبـل أن نمضـي تقـدم أسـمى آيـات الشـكر والامتنـان والتقديروالمحبـة إلـى الـذين حملـوا أقـدس رسـالة فـي الحيـاة إلـى الـذين مهـدوا لنـا طريـق العلـم والمعرفـة ...
إلـى جميـع أساتذتناالأفاضـل...
فواجـب علينـا شـكرهم ووداعهـم ونحـن نخطـو خطوتنـا الأولـى فـي غمارالحيـاة ونخـص بالجزيـل الشـكر والعرفـان إلـى كـل مـن أشـعل شـمعة فـي دروب عملنـا وإلـى مـن وقـف علـى المنـابر وأعطـى مـن حصـيلة فكـره لينيـر دربنـا إلـى الأسـاتذة الكـرام فـي المعهـد العـالى لعلـوم الكمبيـوتر وتكنولوجيـا الاداره بسـوهاج ونتوجـه بالشـكر الجزيـل إلـى

الاستاذ الدكتور

يوســــف بســــيونى مهــــدى

استاذ الحاسبات كلية الحاسبات والمعلومات جامعة اسيوط

الدكتور



استاذ الحاسبات بكلية العلوم بسوهاج

الذين تفضوا بالإشراف على هذا البرنامج فجزاهم الله عنا كل خير فلهم منا كل التقدير والاحترام
وكذلك نشكر كل من ساعد على إتمام هذا البرنامج وقدم لنا العون ومد لنا يد المساعدة وزودنا بالمعلومات اللازمة لإتمام هذا البرنامج ونخص بالذكر:

الاستاذه

تهـــــــانى تمـــــــام عطيـــــــه الدقيشـــــــى

المقدمة-

تطبيق ادارة شئون الطلاب لحل مشكلة الاحتفاظ بمعلومات الطلاب فى السكاشن والقيام بمعالجة العمليات الاساسيه اليوميه والسنويه فى اقسام شيئون الطلاب مثل
(الدرجات –نسبة الحضور- تقييم الطالب)
النظام يمكن ان يعمل متكاملا مع نظام الحضور واعمال السنه
يتيح برنامج المستخدم اكواد وبيانات جميع مكونات واجزاء النظام مثل:-

١. بيانات للطالب:-
(رقم الطالب/الاسم/تاريخ الميلاد/العنوان/الحاله/الفرقه/البريد الالكترونى)
٢. حضور الطلاب فى السكاشن:-
(رقم المعيد/رقم الطالب /رقم القاعه/رقم الماده/تاريخ السكشن/الميعاد/الفرقه /الترم)

يتم ادخل البيانات من خلال نمازج واضحه تحتوى على وسائل تحكم وتحتوى على قوائم جاهزه للاختيار منها بكل سلاسه ويسر.

تحليل النظام:-

تعريف المشكله

ان المشكله الحقيقيه فى نظام شيئون الطلاب فى مؤسسه تستخدم الطريقه اليدويه فى تنفيز جميع الاعمال والمهام المتعلقه بالطلاب يؤدى الى ظهور الكثير من المشاكل ومنها:-

- ظهور بعض الاخطاء فى السجلات والكشوفات اثناء تنفيز الاعمال والمهام المختلفه
- صعوبة متابعة حالة الطالب طوال فترة انتظام الدراسه من حيث (الغياب/الانقطاع/الفصل)
- عدم توفرالسلامه والامان اللازمه للبيانات واحتمال تعرضها للتلف اوالحريق او السرقه فى وقت
- صعوبة استرجاء البيانات عن الطلاب والمعيدين لكثرة السجلات وعم ترتيبها بشكل جيد
- ضياع الوقت فى اجراء جميع العمليات الخاصه بالطالب بسبب الروتين واستخدام الاسلوب اليدوى
- صعوبة اجراء عمليات الارشفه وحفظ الوثائق والملفات الخاصه بالطلاب وذلك لكثرة تلك الوثائق

وللتخلص من جميع وعيوب المشاكل النظام اليدوى نقوم بانشاء نظام معلومات يعتمد على الحاسبات والذى من خلاله نسعى الى تحقيق الاهداف التاليه:-

- التقليل من الاخطاء فى المعلومات الخاصه بالطلاب
- سهولة استرجاع البيانات عن الطلاب وتقييم كل طالب
- المحافظه على البيانات والمعلومات فى النظام وحمايته من المخاطر المختلفه
- سهولة اعداد التقارير المختلفه عن الطلاب والمعيدين ومتابعة خالتهم اثناء الدراسه
- السرعه والكفاءه والموثوقيه فى جميع الاعمال والمهام التى يقوم بها النظام
- تسهيل اجراءات الارشفه وحفظ الوثائق
- الحد من عمليات التزوير والتزويف فى الوثائق الخاصه بالطالب وكذلك الحد من التلاعب بالبيانات الخاصه بالطلاب

٢-دراسة الجدوى

١-الهدف من المشروع:-

متابعة الطلاب فى تسجيل الحضور والغياب فى السكاشن ودرجات اعمال السنه

٢-دراسة الجدوى التقنيه:-

تعنى بالدراسه التقنيه وصف المتطلبات ومستلزمات النظام التكنولوجيه بجانبيها المادى والبرمجى

- **الجانب المادى:- hard ware**

من خلال دراستنا لاحتياجات النظام المادى الكفيله لتشغيل النظام بكفاءه وسرعه عاليه وكزلك من خلال دراسة التقنيه المتوفره فى السوق وللتوفيق بين احتياجات ومتطلبات النظام وبين التقنيه المتوفره فى السوق تم تحديد الاحتياجات الماديه التاليه:-
-جهاز حاسب مواصفات فنيه معقوله.

- **الجانب البرمجى:- soft ware**

وايضا من خلال دراستنا للجانب البرمجى اللازم للنظام والتكنولوجية البرمجية المتوفررة فى السوق تم تحديد المتطلبات التالية :-

1. **برنامج Microsoft Visual Studio 6.0**
2. **برنامج مدير ادارة قواعد البيانات Microsoft Office Access 2003**

ومن خلال المعطيات المذكورة اعلى نجد ان النظام ذو جدوى تقنية.

١) من خلال دراستنا لتكاليف متطلبات ومستلزمات وتشغيل النظام وجدنا الاتى:
٢) تكاليف نظام الحاسب وملحقاتة ومكوناتة من الطابعة وغيرها..........
٣) تكاليف تحليل وتصميم وبرمجة النظام .
٤) تكاليف شراء النظام .
٥) تكاليف تشغيل النظام.

وبعد الانتهاء من دراسة تكاليف النظام تم دراسة العوائد التى سيحققها النظام الالى وهى كالتالى:

١) يتضح من اسم النظام واهدافة انة ليس نظام تجارى او استثمارى وانما نظام خدمى بحت يهدف الى تقديم خدماتة بارقى مستوى وبشكل مرضى لجميع المستفيدين من النظام واستخدام الحاسب هو الاسلوب الامثل لتحقيق هذة الاهداف.
٢) تقليل التكاليف الاضافية التى تتحملها الموسسة والمتمثلة فى استخدام عدد كبير من الموظفين فى تنفيذ مهام ادارة شئون الطلاب .
٣) تقليل المبالغ التى تنفق فى اعداد النمازج والتقارير والكشوفات وذلك باعدادها باستخدام الحاسب بسهولة

وبعد الانتهاء من دراسة العوائد والتكاليف ومع المرعاة الاهداف الخدمية التى سيقدمها النظام المقترح نجد ان العوائد التى سيحققها النظام الالى اكبر بكثير من التكاليف وبذلك يكون النظام ذو جدوى اقتصادية.

٣. دراسة الجدوى التشغيليه:-

عند دراستنا عرفنا ان بعض الموظفين يعرفون بعض المعلومات عن نظام التشغيل Windows وبالتالى سنستخدم نظام التشغيل وندوز وذلك لسهولة هذا النظام وايضا لشهرة هذا الناظام وهو كذلك موجود فى السوق المحليه وهناك ايضا مجال واسع فى عقد دورات لمستخدمى النظام فى شئون الطلاب وسواء كان ذلك داخل المؤسسه او فى مراكز خارجه وبذلك نجد ان النظام زو جدوى تشغيليه .

٣-تصميم النظام

تتكون نظام قاعدة البيانات من الجداول الاتيه :-

١) **Dname :-** جدول المعيدين ويشمل الاتى:-
(رقم المعيد/اسم المعيد)
٢) **Present:-** جدول الحضور ويشمل الاتى :-
(رقم المعيد/رقم القاعة/رقم_الطالب/رقم_المادة/تاريخ_السكشن/الميعاد/الفرقه/الترم)
٣) **Rname:-** جدول القاعات ويشمل الاتى:-
(رقم القاعه / اسم اقاعه / المكان)
٤) **Sbname:-** جدول المواد ويشمل الاتى :-
(الكود / اسم الماده / الفرقه)
٥) **Stname :-** جدول الطلاب ويشمل الاتى:-
(رقم الطالب/الاسم/تاريخ الميلاد/العنوان/الحاله/الفرقه/البريد الالكترونى)
٦) **USERMAST :-** جدول المستخدمين ويشمل الاتى :-
(PASSWORD / NAME)

نظام الحضور:-

- تسجيل حضور الطلاب.
- بحث عن عدد مرات حضور الطالب لتقييم اعمال السنه.

فى مرحلة التصميم يتم تحديد عناصر النظام وهى كافة البيانات التى سيتم استخدامها داخل النظام وهى تتضمن مجموعه من الجداول سوف نستعرض خصائصها فى الاجزاء التاليه ومجموعه الجداول هذه تمثل فى الحياه الواقعيه مايسمى بالسجل العام للطلاب وهو عباره عن سجل يحتوى على جميع البيانات الخاصه بالطلاب للتسجيل وحضور وتقييم درجات الطلاب الخاصه بكل طالب ويسجل الطالب فيه بعد حضوره للسكشن واستكمال جميع البيانات الناقصه حيث يعتبر هذا السجل هو الوثيقه الرئيسيه والمرجع الرئيسى لبيانات الطلاب فى النظام الحالى هذا السجل سوف يتم محاكاته عن طريق مجموعه من الجداول والتى كل منها يخزن مجموعه من

الفصل الأول

عند الحديث عن قواعد البيانات، يتعالى صدى الحروف ADO. NET بين أحاديث المبرمجين، فهي الوسيلة المثلى لربط شيفراتهم البرمجية بملفات قواعد البيانات سواء كانت محلية أو على بعد آلاف الكيلومترات. هذا الفصل هو مدخلك المبدئي إلى ADO. NET واستخدام فئاتها، ودعني انوه هنا بان هذا الجزء مختص ببرمجة قواعد البيانات وليس التعامل مع قواعد البيانات بشكل عام، فلن اتحدث عن طريقة بناء الجداول أو ادارة نظم قواعد البيانات، بل سيكون حديثي محصور حول طرق الوصول إلى قواعد البيانات من شيفراتك المصدرية والمكتوبة بـ Visual Basic .NET.

مدخلك إلى ADO. NET

ADO. NET هي مجموعة من الفئات مشمولة في مجال الاسماء System. Data غرضها الوصول إلى مصادر البيانات Data Sources ، والتي تمثل بيانات محفوظة تحت أنظمة قواعد بيانات متعددة الانواع، الاجناس، والاعراق مما يعني قدرتك على الوصول إلى أي قاعدة بيانات مهما كانت الشركة المنتجة لها.
مهما كان نوع مصدر البيانات الذي تتعامل معه، عند استخدام ADO. NET عليك اختيار وضع من وضعين هما: الوضع المتصل Connected Mode ، والوضع المنفصل Disconnected Mode .

في الوضع المتصل يتم الاتصال مع مصدر البيانات وتجري كافة العمليات – افضل ما أوصفها- على الهواء المباشرة، أو كما انك تجري محادثة هاتفية مع احد الاشخاص، فالاتصال مستمر وحي يرزق بين برنامجك وبين مصدر البيانات الأساسي، وأي خلل في مصدر البيانات أو ايقاف مؤقت، سيؤدي إلى خلل برنامجك.

اما في الوضع المنفصل فهو قريب من فكرة صندوق الوارد Inbox ، حيث تأتيك البيانات وتعدل فيها كما تشاء ومن ثم ترسل كافة التعديلات . وهذا ما يحدث مع ADO. NET بالضبط، فبعد قيامك بإجراء اتصال مع مصدر بيانات، ستحصل على البيانات المطلوبة وتنتهي علاقتك بمصدر البيانات الاساسي، وتتعامل مع البيانات – التي عندك - كما تتعامل معه الوضع المتصل. ليس هذا فقط، بل يمكنك تعديلها وتحريرها واجراء كافة جمل الاستعلام SQL عليها.

ميزة عظيمة في الوضع المنفصل يظهر من خلال استقلالية البيانات عن هيئة مصدر البيانات، ماذا يعني هذا؟ يعني انك تستطيع الحصول على بيانات من مصدر بيانات من النوع Microsoft SQL Server® ، و تتعامل مع البيانات وتحريرها، ومن ثم ارسال البيانات إلى مصدر بيانات من النوع Microsoft Access®. كان الغرض الاساسي من تطوير فكرة الوضع المنفصل هو تخفيف الضغط على مصادر البيانات في اجهزة الخادم Servers والتي تستخدم من قبل عدد كبير من العملاء Clients ،

في هذا الفصل سيتمحور حديثي عن الوضع المتصل، حيث يعتبر مفتاحك لاستخدام والتعامل مع مصادر البيانات في الوضع المنفصل .

مزودات البيانات .NET Data Providers

الميزة العظيمة التي تجنيها من استخدام ADO. NET هو استقلالية برنامجك عن الهيئات المختلفة لمصادر البيانات، أي ان نفس البرامج التي استخدمتها لتطوير تطبيقات معتمدة على مصادر بيانات من نوع Microsoft Access® ، ستستخدمها ايضا مع انواع مصادر بيانات اخرى كـ Microsoft FoxPro® و Microsoft SQL Server® و Oracle® ...الخ، دون الحاجة لتغيير شيفراتك المصدرية.

السؤال الذي يطرح نفسه، كيف يمكن لـ ADO. NET من فعل ذلك رغم انها كتبت مرة واحدة فقط، وبالتالي ستكون موجه إلى نوع معين من مصادر البيانات؟ والجواب هو ان ADO. NET في الحقيقة لا تصل مباشرة إلى مصادر البيانات وانما تعتبر واجهة للمبرمج فقط ، حيث انها تستخدم مزودات البيانات .NET Data Provider. كما يوضحه الشكل التالي:



مزودات البيانات .NET Data Provider ما هي الا حلقة وصل بين ADO. NET ومصادر البيانات، فعندما تريد من ADO. NET استخدام مصدر بيانات من النوع البيانات Microsoft Access® عليك توفير مزود .NET Data Provider والخاص بـ Access® اما ان كبر حجم قاعدة البيانات واردت الاعتماد على خادم كـ Microsoft SQL Server®، فكل ما هو مطلوب منك استخدام المزود الخاص بـ Microsoft SQL Server®.

من منظورك كمبرمج، ستستخدم ADO. NET بغض النظر عن نوع مصدر البيانات، ومن منظورك كمصمم للتطبيق، عليك تحديد مزود .NET Data Providers والذي يلائم نوع مصدر البيانات الذي تستخدمه، كما عليك إرفاق كافة ملفات هذا المزود عند توزيع برنامجك. توجد العشرات من مزودات .NET Data Providers تمثل انواع مختلفة من مصادر البيانات، ويتم تحديث اصدارات كل فترات متعددة، عليك البحث دائما عن اخر اخبار المزود الذي تستخدمه في موقع الشركة المنتجة له.

توجد ثلاثة انواع من مزودات البيانات مدعومة في اطار عمل .NET Framework هي

: OLE DB .NET Data Provider

هذا النوع من المزودات يمكنك من استخدام مزودات OLE DB قديمة مبنية على تقنية COM و يفيدك كثيرا ان اردت استخدام نفس مصادر البيانات والمنجزة قبل تقنية Microsoft .NET .

: SQL Server .NET Data Provider

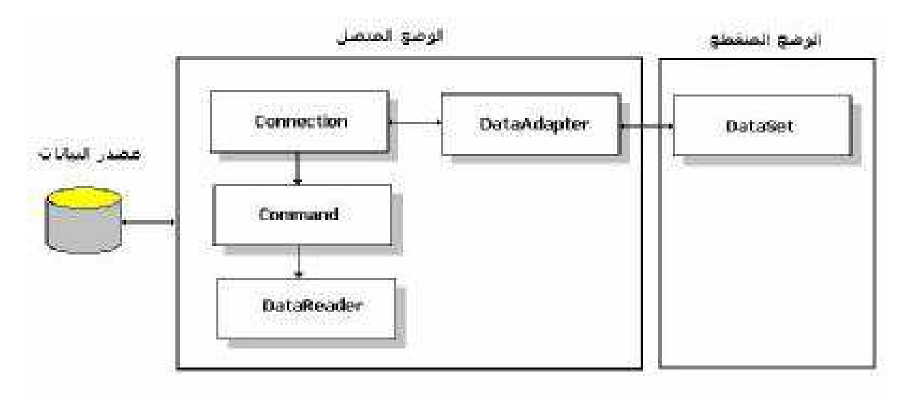
هذا المزود موجه - بشكل خاص- إلى مصادر البيانات المنجزة بالاصدار السابع من Microsoft SQL Server® وما بعده، وان اردت التعامل مع اصدارات اقدم، فعليك استخدام المزودات من النوع OLE DB .NET Data Provider السابقة.

: ODBC .NET Data Provider

قبل تقنية .NET وقبل تقنيات الوصول إلى البيانات، كانت تقنية ODBC اول تقنية تمكن المبرمجين من الوصول إلى مصادر البيانات المختلفة عن طريق مشغلات ODBC Drivers المختلفة (الفكرة من المشغلات والمزودات شبيهه إلى حد كبير من ناحية نظرية). الغرض من المزود ODBC .NET Data Provides هو استخدام مشغلات ODBC Drivers للوصول إلى
مصادر البيانات (وهي اقل كفاءة حتى من مزودات OLE DB .NET Data Providers ولا انصحك بالاعتماد عليها).

فئات ADO. NET

تحتوي ADO. NET على مجموعة كبيرة من الفئات، ولكن اغلبها مشتقة من خمس فئات رئيسية هي انظر شكل (٢): DataSet و DataAdapter و DataReader و Command و Connection وعملية استيعاب هذه الفئات القاعدية سيسهل عليك الامر كثيرا للنزول إلى الفئات المشتقة وفهم طرق استخدامها والربط بين الكائنات المنشئة منها.



شكل (٢) : ابرز فئات الـ ADO. NET.

سأفصل في جميع الفئات - السابق ذكره ا- في هذا الفصل والفصل القادم بمشيئة الله، ولكن دعنا الان ناخذ جولة سريعة حولها . حيث تكون البداية دائما وأبدا مع Connection ، اذ يترتب عليك انشاء كائن من هذه الفئة عن طريقه تتصل بمصدر البيانات، كما ستلحق كل خصائص ومواصفات الاتصال (كاسم المستخدم، كلمة المرور، مزود NET. Data Providerالمستخدم ...الخ) مع هذا الكائن.

بعد انشائك لكائن اتصال، سترسل مرجع لهذا الكائن إلى طرق وخصائص كائن اخر من النوع Command ، يمكنك من تنفيذ جمل الاستعلام المختلفة (كـ SELECT ، DELETE، INSERT وغيرها) على مصدر البيانات والمحدد في الكائنConnection السابق.
بعد تنفيذ جمل الاستعلام في الكائن Command ، ستكون النتائج متمثلة في سجلات Records تصل اليها بين ثنايا الكائن من النوع DataReader لتتمكن من قراءة كل سجل أو كل حقل على حدة، مع العلم انك لن تستطيع تحديث مصدر البيانات من خلال هذا الكائن.

الفئات الثلاث السابق ذكرها تستخدم في حالة الوضع المتصل Connected Mode ، اما الكائنات من النوع DataSetفيوجد الكثير لأخبرك به حولها في الفصل القادم . وحتى ان نلتقي هناك، اعلم انها تمثل البيانات المأخوذة من مصدر البيانات.

أخيرا، يمكنك الحصول على البيان ات من مصدر البيانات أو ارسالها اليه، وإلحاقها إلى الكائنات من النوع DataSetعن طريق كائن من النوع DataAdapter ، يمكنك اعتباره كهمزة الوصل التي تمكنك من ارسال/ استقبال البيانات من والى كائنات Connection.

كائن الاتصال Connection
في عالم ADO. NET ، الاتصال الذي تجريه مع مصدر بيانات هو كائن من النوع Connection . وقبل التعامل مع مصدر بيانات، عليك فتح اتصال معها (حتى ولو كنت في الوضع المنفصل Disconnected Mode) في هذا القسم من الفصل سنعرض الاساليب المتعددة للاتصال بمصادر البيانات.

إنشاء كائن اتصال Connection
عندما تنوي انشاء كائن اتصال فانك لن تقوم بتعريفه من الفئة Connection السابقة وانما ستستخدم نوعين من الاتصال ، النوع الاول للمزودات من النوع Data Provider OLE DB .NET حيث ستعرف كائن اتصال من الفئة OleDbConnection:

```
Dim msAccessCn As New OleDbConnection()
```

اما ان كان مصدر البيانات الذي تنوي الاتصال به يتبع للمزود SQL Server .NET Data Provider ، فالفئة SQLConnection هي المسئولة عن انشاء كائنات من هذا النوع:

```
Dim msSQLCn As New SqlConnection()
```

كلا الفئتين تحتويان على الواجهة IDbConnection مما يمكنك من الاستفادة من مبدأ تعدد الواجهات Polymorphismوتتمكن – مثلا- من كتابة اجراء واحد يستقبل كلا النوعين لاختصار كتابة الشيفرات المكررة:

```
Sub ConnectionDB( ByVal dbCon As IDbConnection)
```

```
...
...
...
End Sub
```

نص الاتصال

بعد انشاء كائن الاتصال، عليك ارفاق نص الاتصال Connection String اليه، ونص الاتصال ما هو إلى قيمة حرفية (من النوع String) تحتوي على كل شيء تتعلق بعملية الاتصال بمصدر البيانات، كاسم مصدر البيانات، مسار قاعدة ملف قاعدة البيانات (أو قد تكون اسم الجهاز في الشبكة ان كنت معتمد على خادم قواعد بيانات كـ Microsoft SQL Server®)، اسم المستخدم، كلمة المرور ... الخ، نص الاتصال التالي مناسب جدا لمصادر بيانات من النوع Microsoft Access® :

```
Dim connString As String = "Provider= Microsoft. Jet. OLEDB. 4.0;" _
& "Data Source= C:\ Folder\ Data. MDB;"
```

حيث Provider هو الاسم الكامل للمزود، و Data Source اسم ملف قاعدة البيانات . المزيد ايضا، يمكنك اسناد وقت الانتظار Connection Timeout بين ثنايا نص الاتصال، ووقت الانتظار هي اطول فترة التي سيظل الكائن منتظرا ردة الفعل من مصدر البيانات عند الاتصال، فنص الاتصال التالي سينتظر ١٠ ثواني لفتح الاتصال مع مصدر البيانات، وان انتهت الفترة ولم تحدث أي ردة فعل من مصدر البيانات نفسه، سيظهر خطأ (القيمة الافتراضية هي ١٥ ثانية):

```
Dim connString As String = "Provider= Microsoft. Jet. OLEDB. 4.0;" _
& "Data Source= C:\ Folder\ Data. MDB; Connection Timeout= 10"
```

بعد كتابتك لنص اتصال، أسنده فورا إلى الخاصية ConnectionString والتابعة لكائن الاتصال، كما يمكنك ارسال نص الاتصال مباشرة إلى مشيد الفئة لحظة انشاء الكائن:

```
Dim msAccessCn As New OleDbConnection()
msAccessCn. ConnectionString = connString

Dim msAccessCn As New OleDbConnection( connString)
```

عندما تتعامل مع الكائنات من النوع SqlConnection ، فعليك تجاهل اسم المزود عند كتابتك لنص الاتصال، والسبب قد يبدوا بديهيا ان علمت ان الكائنات من النوع SqlConnection لا تقبل الا المزودات من النوع SQL Server Data Provider :

```
Dim msSQLCn As New SqlConnection()
msSQLCn. ConnectionString = "Data Source= Asd_ SERVER;" _
& "User ID= اسد; Password= ولا حتى تحلم فيها "
& "Initial Catalog= قاعدة بيانات المقالات"
```

فتح وإغلاق الاتصالات

بعد اسنادك لنص الاتصال المناسب للخاصية ConnectionString ، يمكنك البدء بفتح الاتصال مع مصدر البيانات باستدعاء الطريقة Open():

```
Dim msAccessCn As New OleDbConnection( connString)
Dim msSQLCn As New SqlConnection( connString2)
msAccessCn. Open ()
msSQLCn. Open ()
```

ومن الضروري جدا اغلاق الاتصال عند عدم الحاجة اليه باستدعاء الطريقة ()Close:

```
msAccessCn. Close ()
msSQLCn. Close ()
```

تستطيع معرفة حالة الاتصال عن طريق الخاصية State التابعة لكائن الاتصال Connecting، والتي قد تكون قيمة أو اكثر من القيم : الاتصال مفتوح Open الاتصال مغلق، Closed جاري فتح الاتصال، Executing يتم تنفيذ امر استعلام على الاتصال، و Fetching جاري الحصول على بيانات من سجلات مصدر البيانات:

```
Dim cn As New OleDbConnection()
…
…
…
If (cn.State And ConnectionState.Open) <> 0 Then
cn. Close()
End If
```

المزيد ايضا، عند تغيير حالة الاتصال من Open إلى Closed (أو العكس) سيتم تفجير الحدث StateChange والخاص بكائن الاتصال :

```
Dim WithEvents cn As New OleDbConnection()
Sub cn_ StateChange( ByVal sender As Object, _
ByVal e As System. Data. StateChangeEventArgs)
Handles cn. StateChange
If (e. CurrentState And ConnectionState. Open) <> 0 Then
Label1.Text = "تم فتح الاتصال"
Elseif e.CurrentState = ConnectionState.Closed Then
Label1.Text = "تم اغلاق الاتصال"
End If
End Sub
```

تفادي الاستثناءات:

عند التعامل مع مصادر البيانات، فان نسبة وقوع الاستثناءات كبيرة جدا لأي سبب أو خلل فني، لذلك ينصح بشدة من تفادي الاستثناءات ووضع الشيفرات الخاصة بمصادر البيانات داخل التركيب Try … End Try :

```
Dim cn As New OleDbConnection()
Try
cn. Open()
…
…
…
Catch ex As Exception
MsgBox( ex. Message)
End Try
```

بالنسبة للطريقة ()Close السابقة، عليك استدعائها دائما لتغلق الاتصال، حيث ان Garbage الاتصالات لا يتم اغلاقها الا لحظة الموت الحقيقي للكائنات (عندما تبدأ المجموعة عملها) ، وبما اننا لا نعلم متى سيحدث هذا فعلينا استدعائها دائما، سواء وقع استثناءCollection أو لم يقع:

```
Dim cn As New OleDbConnection()
```

```
Try
cn. Open()
…
…
…
Catch ex As Exception
MsgBox( ex. Message)
Finally
cn. Close()
End Try
```

كما يفضل الاعتماد على كائنات الاستثناء ات OleDbException أو SqlException للتفريـق بين الاستثناءات الخاصة بكائنات الاتصالات والاستثناءات الاخرى:

```
Try
…
…
Catch ex As OleDbException
…
…
Catch ex As Exception
…
…
End Try
```

كائن الأوامر Command

بعد تكوين الاتصال مع قاعدة البيانات، تأتي الخطوة التالية وهي ارسال جمل الاستعلام SQL إلى قاعدة البيانات لتعديل محتوياتها، في هذا القسم سنرى كيف يمكنك الاستفادة من كـائن الاتـصال Command.وتعديل بيانات قاعدة البيانات عن طريق كائن الاوامرConnection .

<u>إنشاء كائن أوامر</u>

عندما تنوي انشاء كائن اوامر Command فانك لن تقوم بتعريفه من الفئة Command ، وانمـا ستستخدم نوع يماثل مزود كائن الاتصال Connection ، فان كان مزود كائن الاتصال من النوع OLE DB .NET Data Provider ، ستعرف كائن اوامر من الفئة OleDbCommand:

```
Dim msAccessCmd As New OleDbCommand ()
```

اما ان كان مزود كائن الاتصال الذي تنوي استخدامه يتبع للمزود SQL Server .NET Data ، فالفئة SqlCommand هي المسئولة عن انشاء كائنات اوامر خاصة لها :

```
Dim msSQLCmb As New SqlCommand()
```

كلا الفئتين تحتويان على الواجهة IDbCommand مما يمكنك من الاستفادة من مبدأ تعدد الواجهات Polymorphismوتتمكن – مثلا- من كتابة إجراء واحد يستقبل كلا النوعين للتقليص من عدد الشيفرات المكررة:

```
Sub CommandDB( ByVal dbCmd As IDbCommand)
…
…
…
End Sub
```

الربط مع اتصال
بعد إنشائك لكائن اوامر Command ، اول خطوة عليك إنجازها هي ربطه مع كائن اتصال Connection:، يمكنك اسناد مرجع إلى كائن اتصال عن طريق الخاصية Connection

```
Dim cn As New OleDbConnection( connString)
Dim cmd As New OleDbCommand()
cn.Open()
cmd.Connection = cn
```

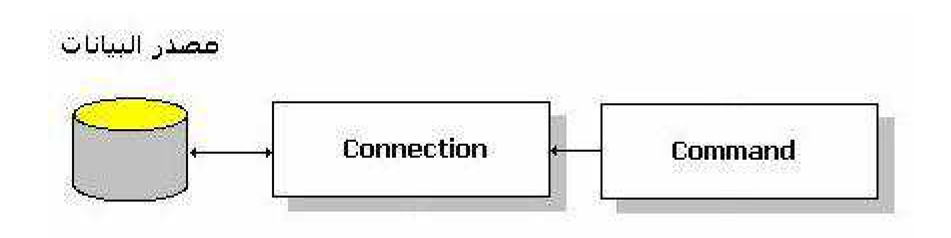
ولا تنسى ضرورة توافق نوع كائن الاوامر مع نوع كائن الاتصال، فلو كان كائن الاتصال يتبع مزود من النوع SQL .NET Data Provider فقد تم إنشائه من الفئة SqlConnection ، لذلك عليك الاعتماد على الفئة SqlCommand عوضا عن OLEDbCommand للربط الصحيح مع الاتصال:

```
Dim cn As New SqlConnection( connString)
Dim cmd As New SqlCommand()
cn.Open()
cmd.Connection = cn
```

ضع في عين الاعتبار، ان كائن الاوامر Command مرتبط ارتباطا وثيقا بكائن الاتصال Connection، فلو تم – مثلا- اغلاق كائن الاتصال بالطريقة Close() فلن تتمكن من تنفيذ جمل الاستعلام SQL مع كائن الاوامر:

```
Dim cn As New OleDbConnection( connString)
Dim cmd As New OleDbCommand()
cn.Open()
cmd.Connection = cn
cn. Close()
cmd.ExecuteNonQuery()
```

سيظهر لك السبب واضحا ان علمت ان كائن الاوامر لا يصل إلى مصدر البيانات بشكل مباشر، وانما يعتمد على كائن اتصال والذي بدوره يصل إلى مصدر بيانات .



تنفيذ جمل الاستعلام SQL
ان اردت تنفيذ جمل استعلام على الكائن Command ، فعليك تحديد نوع جملة الاستعلام SQL التي تود تنفيذه، هل هي جملة استعلامية تقليدية أو جملة تنفيذية؟

الجمل الاستعلامية هي تلك الجمل التي لا تؤثر على سجلات قاعدة البيانات وانما تقوم بقراءة محتوياتها، في لغة الاستعلام SQL نستخدم الامر SELECT لهذا النوع من الجمل، وسترى في الفقرة التالية قراءة السجلات كيف يمكنك تنفيذها على كائن Command باستدعاء الطرق ExecuteXXX().

اما ان كانت جمل الاستعلام SQL هي جمل تنفيذية، عليك استخدام الطريقة ExecuteNonQuery() والتي تعود بقيمة عددية تمثل عدد السجلات التي تأثرت. الجمل التنفيذية –التي اقصدها في هذا السياق - هي تلك الجمل التي تحدث تغييرا في سجلات جداول قاعدة البيانات مثل DELETE أو INSERT INTO ، UPDATE.

حتى تتمكن من استدعاء الطريقـةExecuteNonQuery() عليـك اولا كتابـة جملة الاستعلام SQL فى الخاصية CommandText والتابعة للكائن Command:

```
Dim cn As New OleDbConnection( connString)
cn. Open()
Dim sqlStatement As String = "UPDATE    asd  set name='asd' where ID=2000"
Dim cmd As New OleDbCommand()
cmd.Connection = cn
cmd.CommandText = sqlStatement
cmd.ExecuteNonQuery()
```

يمكنك دمج السطرين والمتعلقين بتعديل قيم الخاصيتين Connection و CommandText فـي Command: سطر واحد، وذلك بارسال امر جملة الاستعلام وكائن الاتصال مع مشيد الفئة

```
Dim cmd As New OleDbCommand ( sqlStatement, cn)
```

اخيرا، دعني اعيد تذكيرك بضرورة تفادي الاستثناءات لحظة تنفيذ جمل الاسـتعلام – ففد تتغير صلاحياتك على مصدر البيانات أو يحدث أي خلل فني في عملية التحديث :

```
Try
cmd. ExecuteNonQuery ()
…
…
Catch ex As Exception
…
…
End Try
```

قراءة السجلات
عندما تنوي استخدام الامر SELECT في جمل الاستعلام لقراءة السجلات، فيمكن اختيار طريقة من ثلاث طرق تابعة للكائن Command هـي: ExecuteReader() و ExecuteScalar() و ExecuteXMLReader .

الطريقة ExecuteReader():
الطريقة ExecuteReader() تعود بكائن من النوع DataReader تمثل نتيجة جملة الاستعلام في الكائن (Commandسأتحدث عن الكائن DataReader لاحقـا فـي القسم كـائن البيانـات DataReaderمن هذا الفصل):

```
Dim cmd As New OleDbCommand(" SELECT * FROM asd, cn")
Dim dr As OleDbDataReader = cmd.ExecuteReader()
Do While dr.Read()
 -------------------------------
------------------------------
Loop
dr.Close()
```

الطريقة ExecuteScalar():

تستخدم الطريقة ExecuteScalar() لقراءة حقل Field واحد فقط من حقول السجل، وتعود بقيمة تمثل ذلك الحقل:

```
Dim cmd As New OleDbCommand(" SELECT  name from asd where id=10", cn)
Dim Name As String = cmd.ExecuteScalar()
```

تفديك الطريقة ExecuteScalar() كثيرا ان اردت قراءة قيمة واحدة من من السجل وذلك لزيادة سرعة التنفيذ، ان كانت جملة الاستعلام تعود باكثر من حقل، فالحقل الاول هو الذي سيتم قراءته، وان كانت جملة الاستعلام تعود بأكثر من سجل، فسيتم قراءة الحقل التابع للسجل الاول.
يمكنك الاستفادة من الطريقة ExecuteScalar() في معرفة عدد السجلات بشكل سريع– على سبيل المثال لا الحصر:

```
Dim cmd As New OleDbCommand(" SELECT COUNT(*) FROM asd ", cn)
Dim countOfFrogs As Integer = CInt( cmd. ExecuteScalar())
```

الطريقة ExecuteXMLReader():
ان استخدم المزود SQL Server .NET Data Provider ، فستتمكن من استدعاء الطريقة ExecuteXMLReader()، حيث ان قواعد البيانات المعتمدة على انظمة Microsoft SQL يمكن استخدام يمكن استخدام الامر FOR XML من اوامر لغة الاستعلامSQL معها، والتي تعود بالسجلات بهيئة XML .
اذا كانت الطريقة ExecuteReader() تعود بكائن من النوع DataReader ، فان الطريقة ExecuteXMLReader() تعود بكائن من النوع System.XML.XmlReader واستخدامها شبيه – إلى حد كبير- مع DataReader.

```
Dim cmd As New SqlCommand(" SELECT * FROM asd ",cn)
Dim xmlr As System. Xml.XmlReader = cmd. ExecuteXmlReader()
Do While xmlr.Read()
MsgBox( xmlr.Value)
Loop
xmlr.Close()
```

كائن البيانات DataReader
بعد تنفيذ جملة الاستعلام في الكائن Command باستدعاء الطريقة السابقة ExecuteReader() ، ستعود هذه الطريقة بكائن بيانات من النوع DataReader يمثل جميع السجلات الناتجة من جملة الاستعلام، وقبل ان نرى كيف يمكنك الاستفادة من هذا الكائن، من الجيد معرفة كيف يمكنك الحصول عليه – أي إنشائه.

إنشاء كائن بيانات
عندما تنوي إنشاء كائن بيانات DataReader فانك لن تستطيع استخدام الامر New ، وانما ستضطر إلى استخدام الطريقة ExecuteReader() والتابعة للكائن Command ، فان كان مزود الكائن Command من النوع OLE DB ، فانك ستعرف كائن بيانات من الفئة OleDbDataReader:

```
Dim cmd As New OleDbCommand(" SELECT * FROM asd", cn)
Dim dr As OleDbDataReader = cmd. ExecuteReader()
```

اما ان كان مزود كائن الاوامر الذي تنوي استخدامه يتبع للمزود SQL Server فالفئة SqlDataReader هي المسئولة عن إنشاء كائنات بيانات خاصة .

```
Dim cmd As New SqlCommand (" SELECT * FROM  asd", cn)
Dim dr As SqlDataReader = cmd.ExecuteReader
```

مرة اخرى، الواجهة IDataReader مشمولة في كلا الفئتين OleDbDataReader و SqlDataReader ، فيمكنك تعريف اجراء يستقبل في وسيطته كائن من كلا النوعين :

```
Sub ReaderDB( ByVal dr As IDataReader)
…
End Sub
```

قراءة السجلات

بعد إنشائك لكائن بيانات DataReader ، تستطيع البدء في قراءة سجلاته في خطوتين، الاولى باستدعاء الطريقة Read() لتحميل حقول السجل، والثانية تتم فيها قراءة قيمة الخاصية Item والتي ترسل معها اسم الحقل المراد قراءته:

```
Dim dr As OleDbDataReader = cmd. ExecuteReader()
dr. Read()
MsgBox ( dr. Item(" -------“)
dr. Read()
MsgBox dr. Item(" -------“)
dr. Close()
```

في كل مرة تستدعي فيها الطريقة Read() سيتم نقل المؤشر إلى السجل التالي، وستعود الطريقة بالقيمة False ان وصلت إلى نهاية السجلات، لذلك الاستخدام الامثل لها يكون في حلقة بهذا الشكل Do … Loop :

```
Dim dr As OleDbDataReader = cmd. ExecuteReader()
…
…
Do While dr. Read()
Msgbox (dr. Item(" ------")
Loop
dr.Close()
```

الخاصية Item تعود دائما بقيمة من النوع Object ، مما يضطرك إلى الاعتماد على دوال التحويل CInt() ، CLng()، ... CSng() الخ) لقراءة القيمة، مع ذلك يمكنك الاعتماد على الطرق Getxxx() التي تعود بالنوع المكافئ – دون الحاجة لاستخدام دوال التحويل:

```
Dim age As Integer
age = CInt( dr. Item("---")
age = dr.GetInt32( 1)
```

من المهم التنبيه هنا بضرورة اغلاق كائن DataReader باستدعاء طريقته Close() ، السبب في ذلك ليس فقط من اجل تحرير مصادر النظام، وانما يتعدى ذلك بكثير، اذ بمجرد قيامك Connection، سيتم شلل كافة العمليات الاخرى على كائن الاتصالDataReader بإنشاء كائن والاوامر Command، ولن تتمكن من عمل أي شيء الا استدعاء الطريقة Close() لكائن الاتصال في هذه الحالة.

اخيرا، تحتوي الكائنات من النوع DataReader على مجموعة اضافية من الطرق والخصائص المفيدة كالخاصية GetNameالتي تعود باسم الحقل (وليس قيمته) ، الخاصية FieldCounter التي تعود بعدد الحقول التي تعود بالقيمة والاهم من ذلك الطريقة IsDBNull والتى تعود بالقيمة True ان كان الحقل فارغ Null .

وهنا يتم انشاء قاعدتين للبيانات هما Registration, Contacts

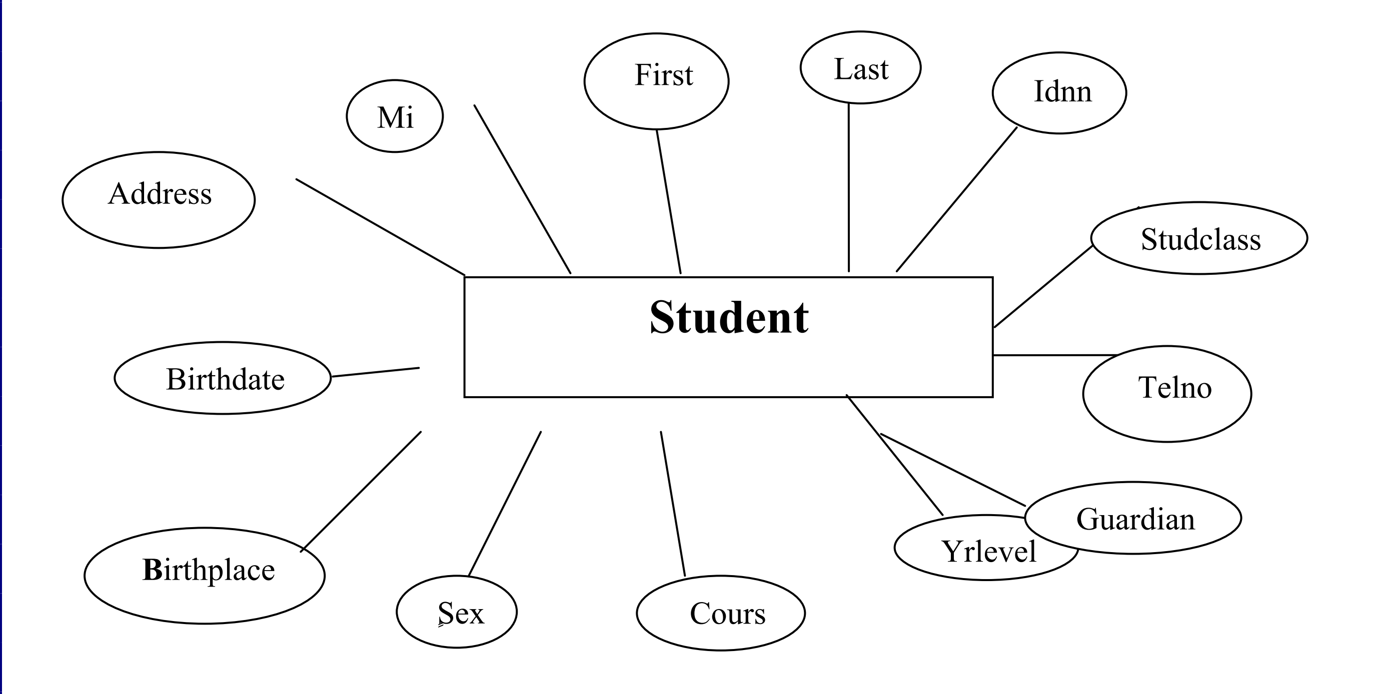
اولا: قاعدة البيانات Registration

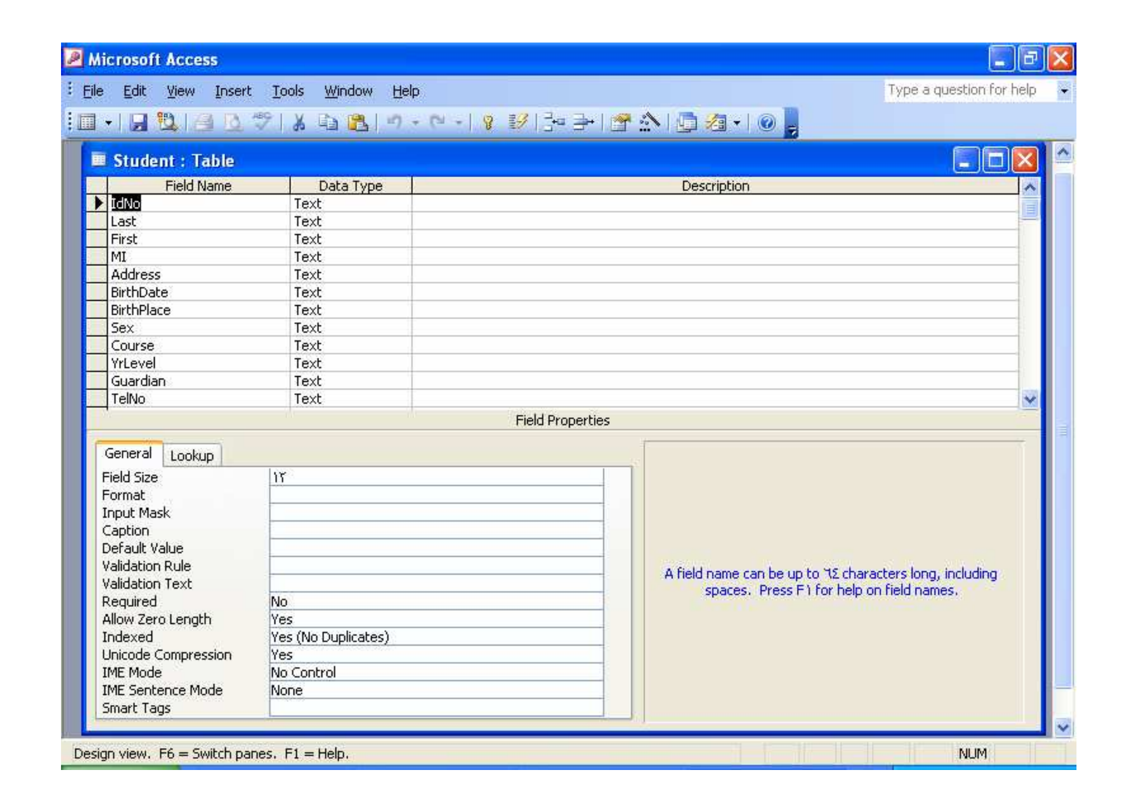
يتكون من جدولين

م	اسم الحقل	النوع	ملاحظات
٢	Last	text	مطلوب
٣	First	text	مطلوب
٤	MI	Text	مطلب
	Address	text	مطلوب
	IdNo	Text	مطلوب
٦	BirthDate	text	مطلوب

١-جدول student	٧	BirthPlace	text	مطلوب
	٨	Sex	text	مطلوب
	٩	Course	text	مطلوب
	١٠	YrLevel	text	مطلوب
	١١	Guardian	text	مطلوب
	١٢	TelNo	text	مطلوب
	١٣	StudClass	text	مطلوب
	١٤	Trimester	text	مطلوب
	١٥	Sy_start	text	مطلوب
	١٦	Sy_end	text	مطلوب

مخطط الكينونه والعلاقه E-R diagram

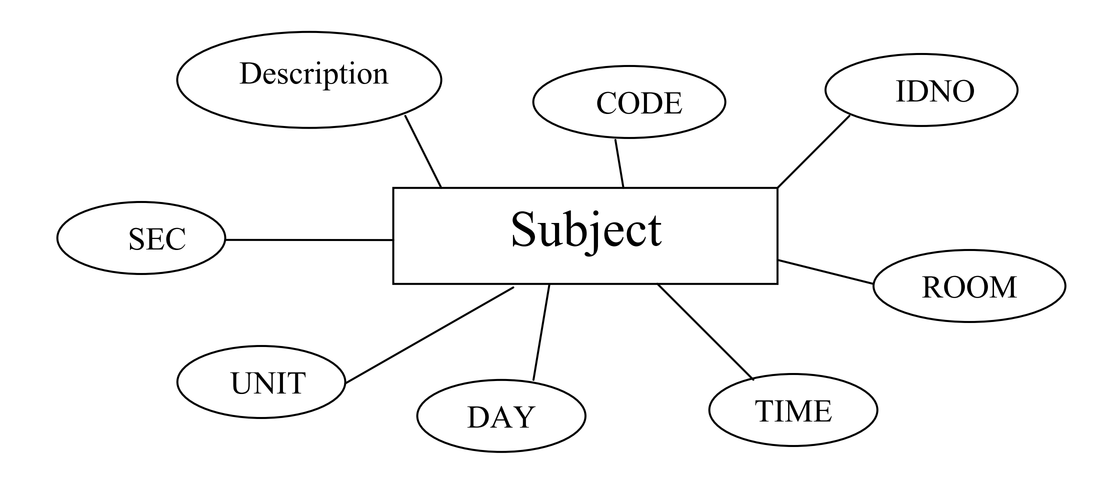


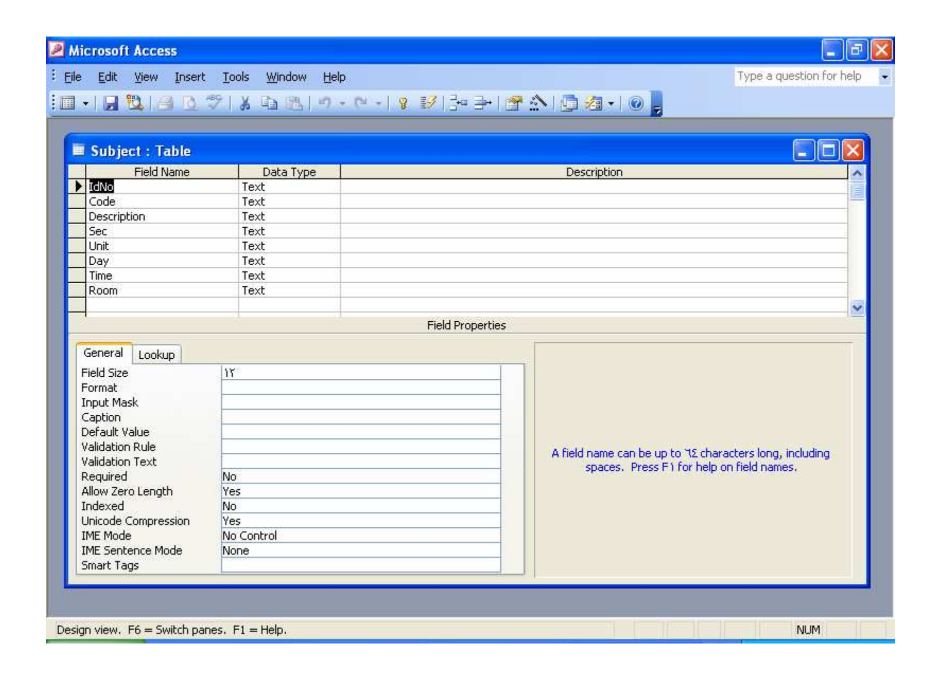


٢- جدول Subjects

م	اسم الحقل	النوع	الملاحظات
١	IdNo	Text	مطلوب
٢	Code	Text	مطلوب
٣	Description	Text	مطلوب
٤	Sec	Text	مطلوب
٥	Unit	Text	مطلوب
٦	Day	Text	مطلوب
٧	Time	Text	مطلوب
٨	Room	Text	مطلوب

مخطط الكينونة والعلاقة



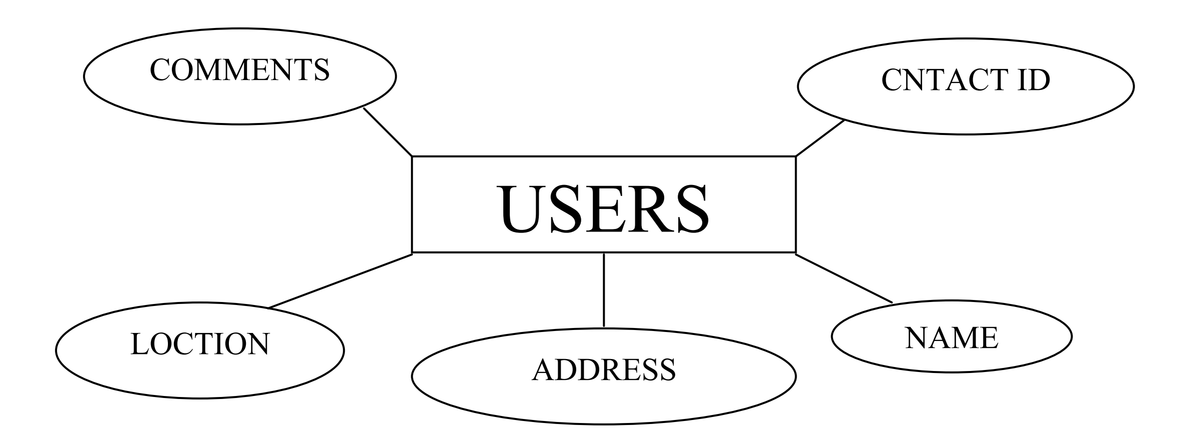

Microsoft Access
File Edit View Insert Tools Window Help
Type a question for help
Subject : Table
Field Name
Data Type
Description
IdNo Text
Code Text
Description Text
Sec Text
Unit Text
Day Text
Time Text
Room Text
Field Properties
General
Lookup
Field Size ١٢
Format
Input Mask
Caption
Default Value
Validation Rule
Validation Text
Required No
Allow Zero Length Yes
Indexed No
Unicode Compression Yes
IME Mode No Control
IME Sentence Mode None
Smart Tags
A field name can be up to ٦٤ characters long, including spaces. Press F١ for help on field names.
Design view. F6 = Switch panes. F1 = Help.
NUM

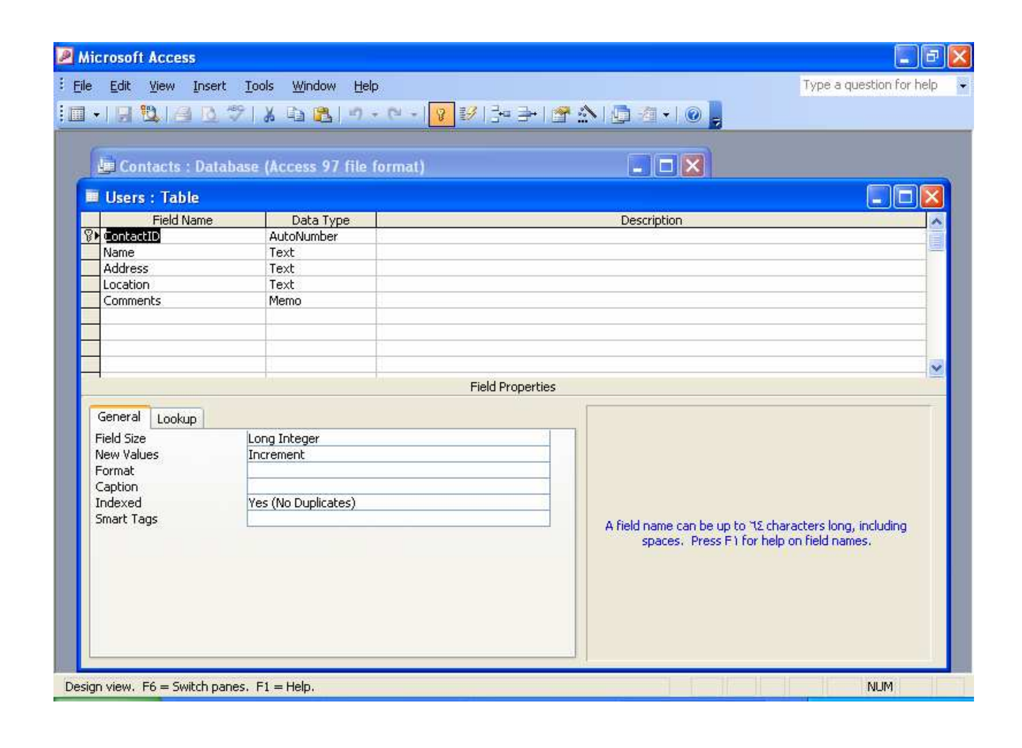
ثانيا: قاعدة البيانات Contacts

1- جدول Users

م	اسم الحقل	النوع	ملاحظات
١	CONTACT ID	رقم	مطلوب
٢	NAME	نص	مطلوب
٣	ADDRESS	نص	مطلوب
٤	LOCATION	نص	مطلوب
٥	COMMENTS	مذكرة	مطلوب

مخطط الكينونة والعلاقة E-R-Diagram





الفصل الثانى

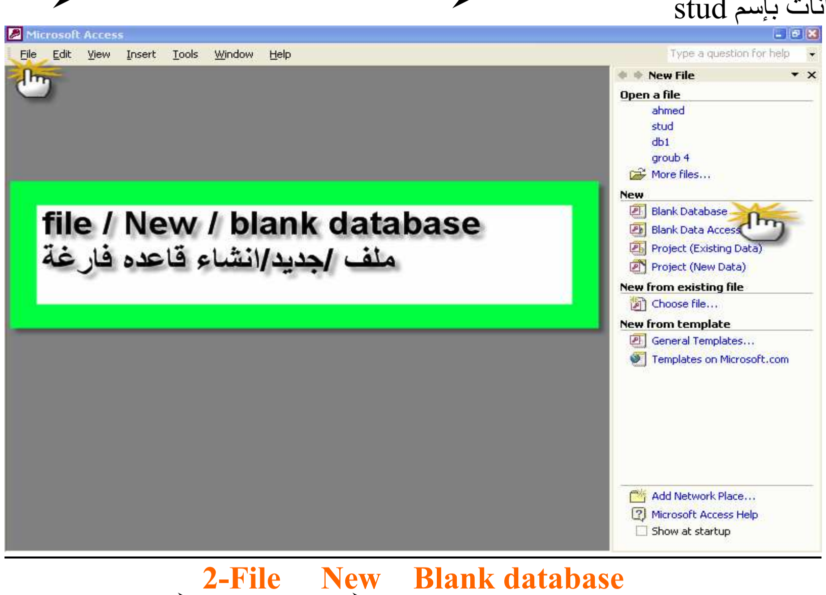
قواعد البيانات لبرنامج متابعة حضور الطلاب السكاشن

تتكون قاعدة البيانات لبرنامج متابعة حضور الطلاب السكاشن من عدد ٦ جدول كالاتى:(بيانات الطلاب – بيانات المعيدين –بيانات القاعات – بيانات المواد الدراسية – جدول الحضور - جدول المستخدمين)

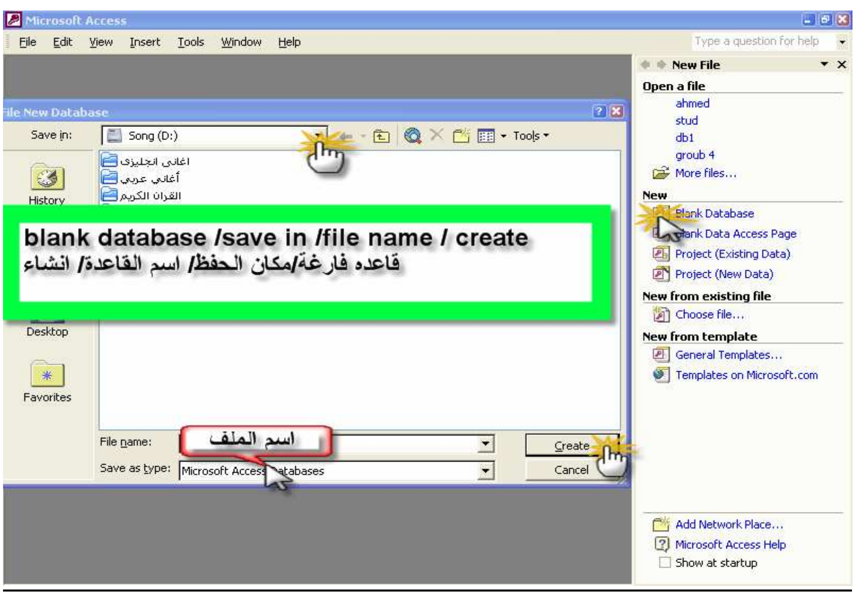
- تم استخدام برنامج اكسيس ٢٠٠٣ لعمل قاعدة بيانات وبها الجداول المستخدمة وهم :

1-start → all programs → Microsoft Access

تم انشاء قاعدة بيانات بإسم stud

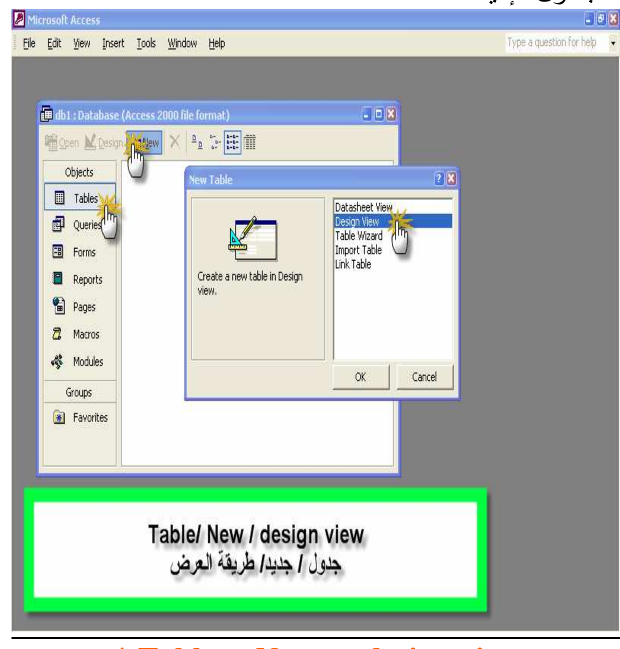


2-File → New → Blank database

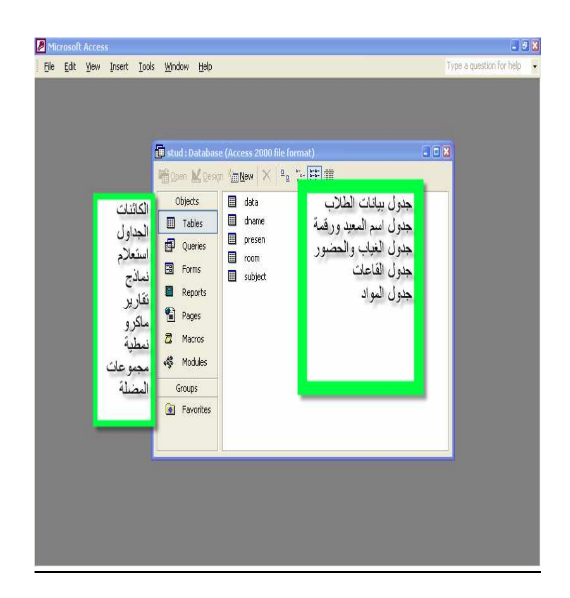


3-Blank → database → Save in → file name → Create →

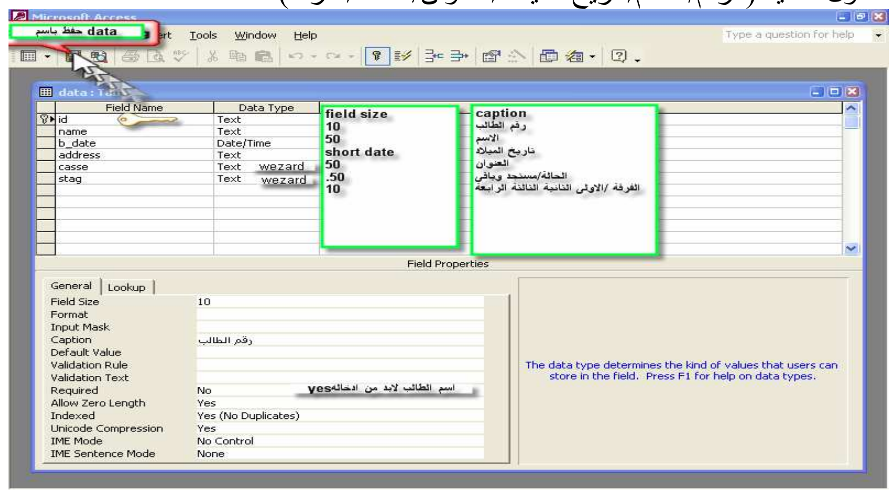
اشتملت قاعدة البيانات على الجداول الإتية



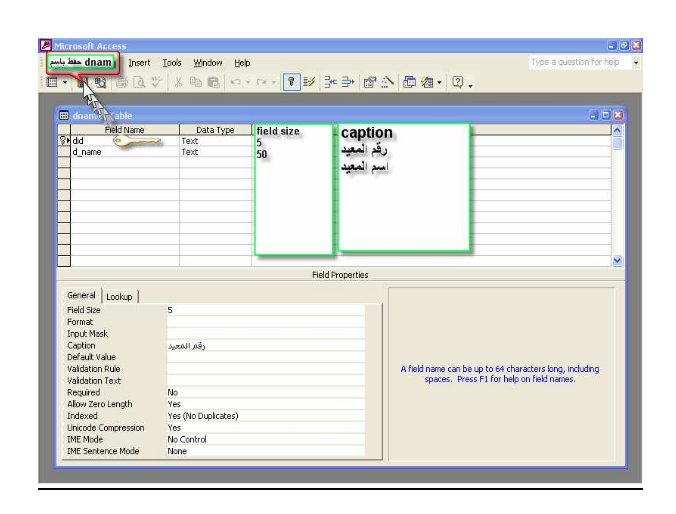
4-Table → New → design view



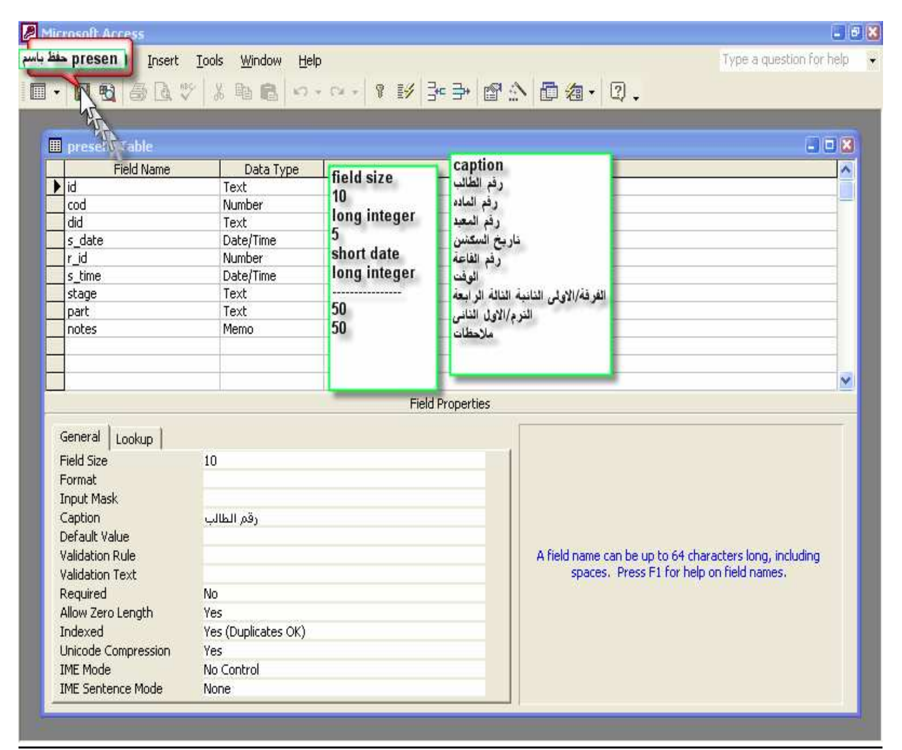
١- جدول بيانات الطالب (stname)
ويشتمل على الحقول التالية (الرقم/الاسم/تاريخ الميلاد/العنوان/الحالة/الفرقة)



٢- جدول بيانات المعيدين(dname)
ويشتمل على (رقم المعيد/اسم المعيد)

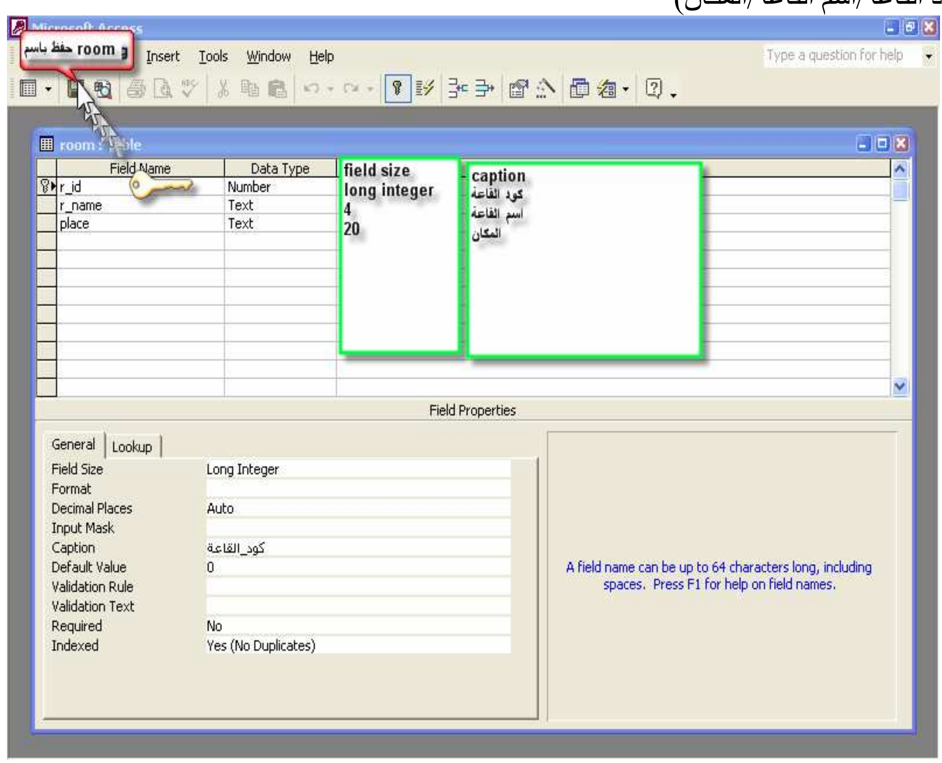


٣- جدول الغياب والحضور(present)
ويشمل (رقم الطالب/رقم الماده/رقم المعيد/تاريخ السكشن/رقم القاعة/الوقت/الفرقة/الترم/ملاحظات)



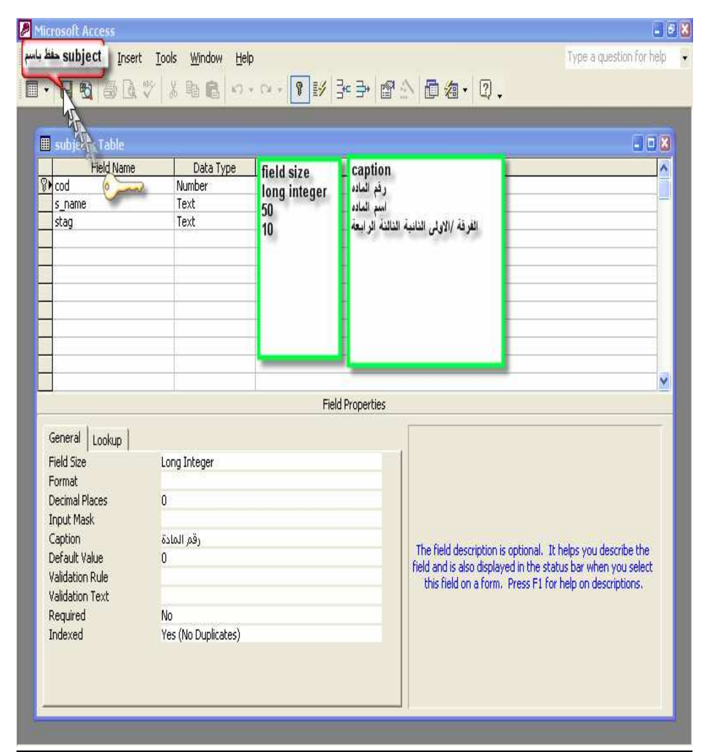
٤- جدول القاعات:(rname)

ويشمل(كود القاعة/اسم القاعة/المكان)



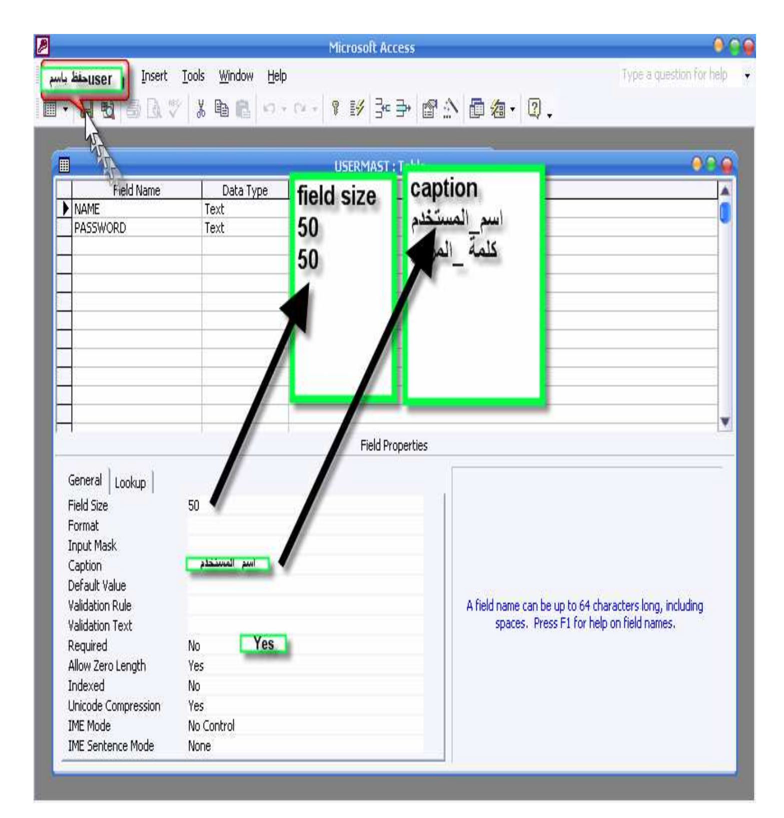
٥- جدول المواد (sbname)

ويشمل(رقم الماده/اسم المادة/الفرقة)



٦- جدول التسجيل (USERMAST)

ويشمل (اسم المستخدم كلمة المرور)



الفصل الثالث





١ ـ نموذج التحميل الخاص بأسم المستخدم وكلمة المرور

الاكواد

```
Private Sub Command1_Click()
Unload Me
End Sub
```

```
Private Sub login_Click()
If txtpassword.Text = IIf(IsNull(Adodc1.Recordset.Fields("password")), "", Adodc1.Recordset.Fields("password")) And
txtusername.Text = Adodc1.Recordset.Fields("name") Then
MsgBox "Login succeded"

Load MDIForm1
Unload Me
MDIForm1.Show
Beep
Beep
Else
MsgBox "wrong password Try again"
txtpassword.Text = ""
txtpassword.SetFocus
End If
End Sub
```

أمر زر الخروج & وأمر زر التحميل من القاعدة



٢ـ النموذج الرئيسى (الشاشة الافتتاحية)

الكود

```
Private Sub Form1_Click()
End Sub

Private Sub aaa_Click()
Form6.Show
End Sub

Private Sub md_Click()
Form1.Show
End Sub

Private Sub mstud_Click()
Form3.Show
End Sub

Private Sub Picture1_Click(Index As Integer)
'frmIssue.Show
End Sub

Private Sub pp_Click()
Form5.Show
End Sub

Private Sub pres_Click()
frmIssue.Show
End Sub
```

باقى الكود

```
Private Sub room_Click()
Form2.Show
End Sub

Private Sub sub_Click()
Form4.Show
End Sub

Private Sub us_Click()
usermaster.Show
End Sub

Private Sub zs_Click()
search.Show
End Sub
```

٣-هذا الكود خاص بالقوائم عند الضغط على الفورم او النموذج يعمل اظهار **show**

عند الضغط على اسم النموذج مثلا القاعات الامر يعمل **show** اظهار الفورم



٤-نموذج بيانات المعيدين



٥- نموذج بيانات الطلاب



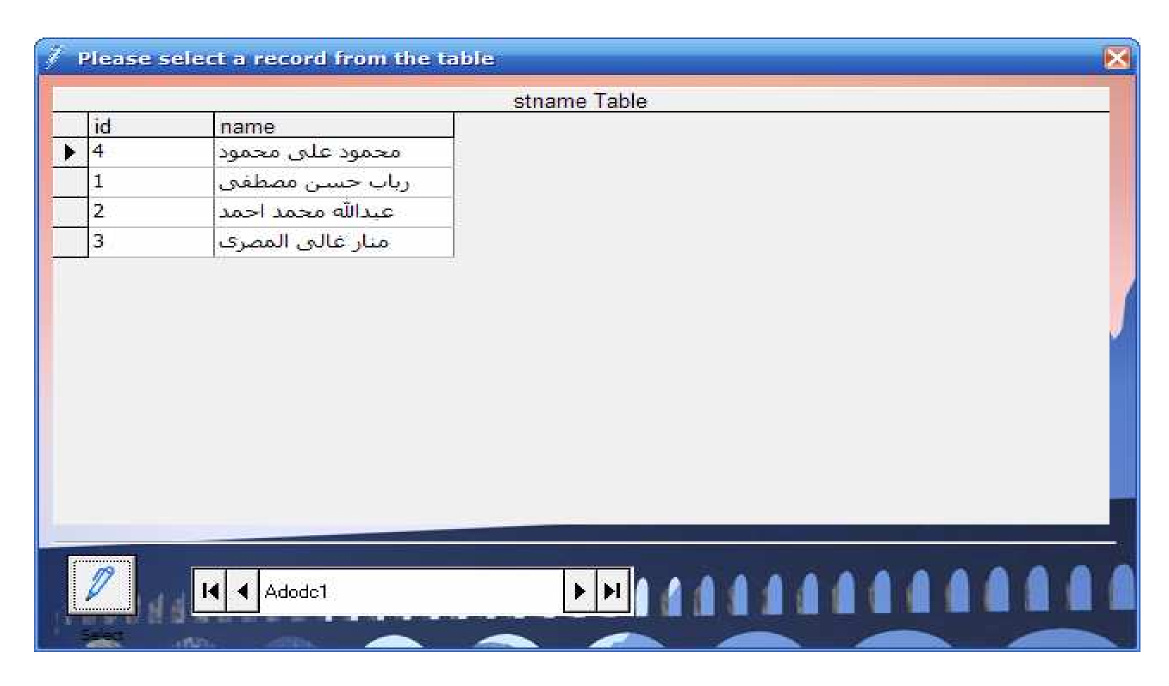
٦_ نموذج بيانات القاعات



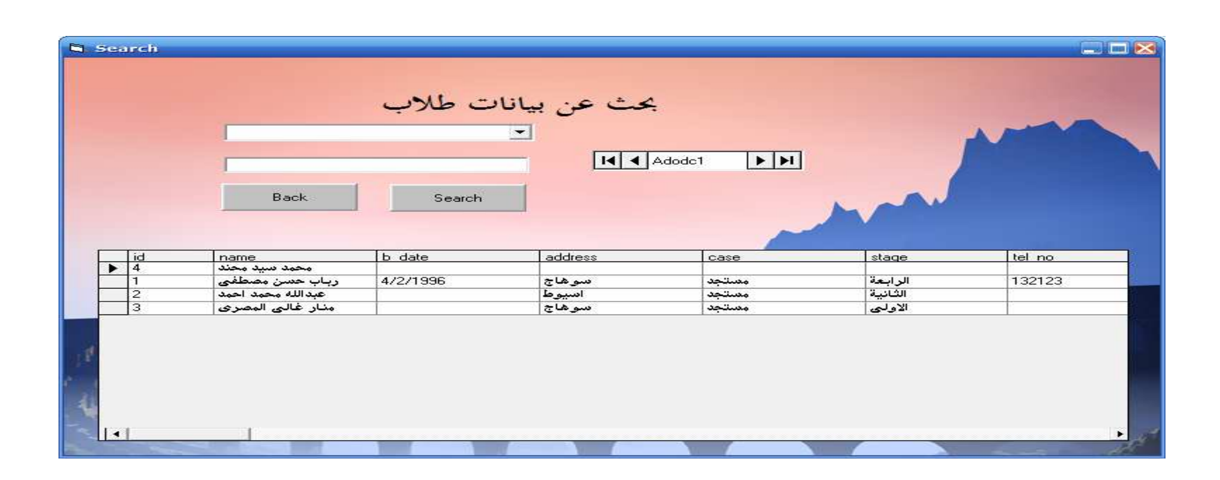
٧- نموذج بيانات المواد الدراسية



٨- نموذج بيانات تسجيل الحضور



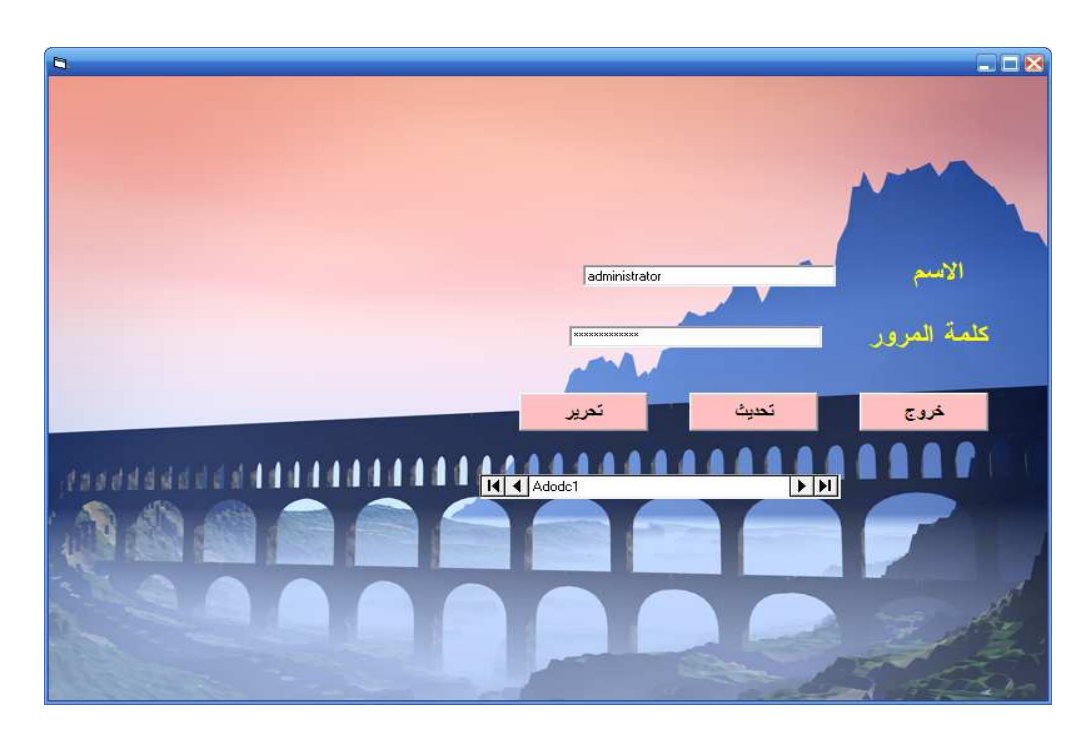
٩ـ نموذج الدخول على بيانات الطلاب لاختيار الاسم او الرقم المطلوب



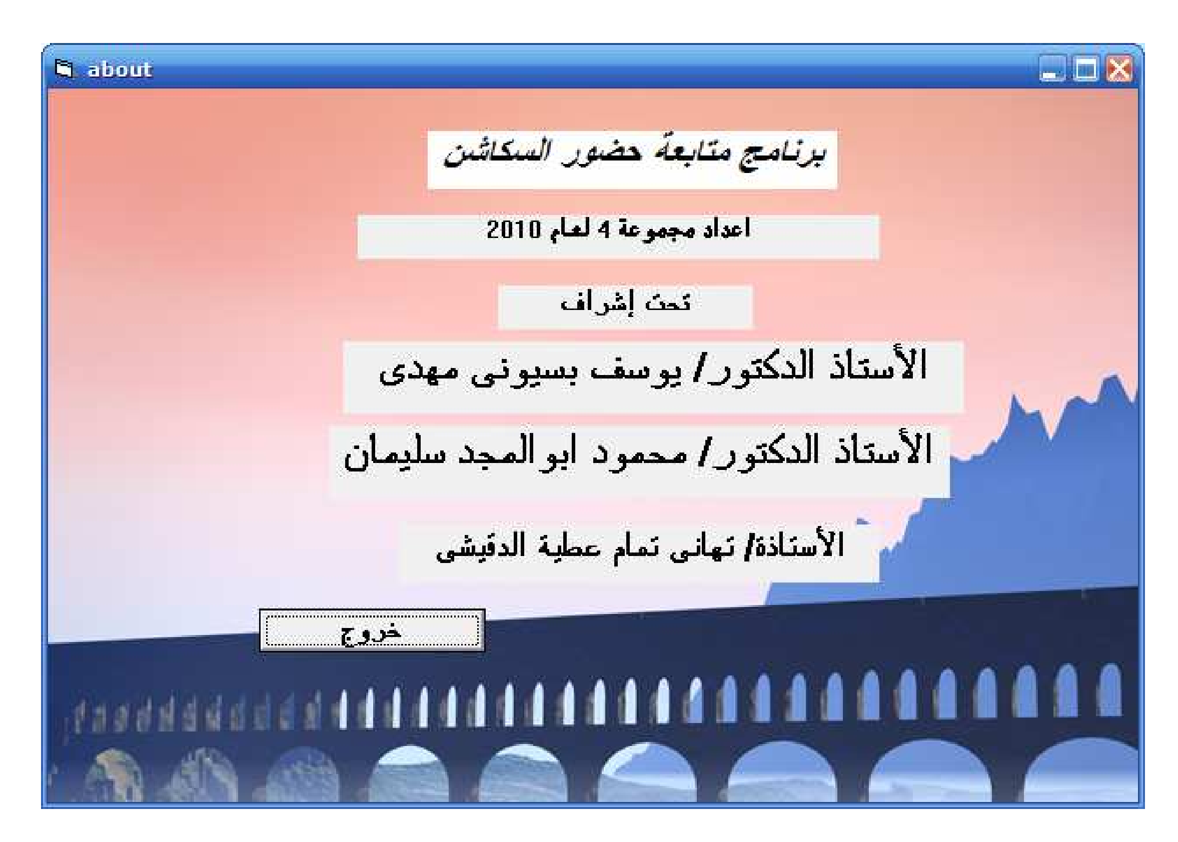
١٠ـ نموذج البحث عن بيانات الطلاب



١١- نموذج بحث عن الحضور



١٢- نموذج تسجيل مستخدم جديد



١٣- **نموذج** عن معلومات الدعم الفنى

اعداد /أحمد فتحى محمد سليمان الشريف الجلوى
محافظة سوهاج مركز ساقلتة الجلاوية
E_mail:tv22000@yahoo.com
Tel:002/0107976725



